AEA by Ivory & Co.

Environment Architecture Standard

Version: v1.0.0 — Published Anchor: Enablement

1. Definition

> Environment Architecture defines the physical and technological enablement required to execute the enterprise's instructions for delivering its intended outcomes.

Environment Architecture is the architecture of Enablement.

Its governing question is:

> What physical and technological enablement must be provided, and how will that requirement be fulfilled, for the enterprise's instructions to be executable as intended?

Environment Architecture owns both the Environment Requirement and the Requirement Fulfilment through which that enablement is provided.

It does not own the work being performed, the Participant performing it, or the actual condition of the Environment at a point in time.

2. Why Environment Architecture Exists

Enterprise work requires conditions in which it can occur.

A process may require:

- a physical workplace;
- machinery;
- equipment;
- a device;
- an application;
- infrastructure;
- connectivity;
- a platform;
- an agent runtime;
- storage;
- computing capacity;
- security mechanisms;
- or another physical or technological resource.

Execution can define the work without providing those conditions.

Workforce can define participation without providing them.

Environment Architecture exists to make the required physical and technological enablement explicit and to establish how those requirements are fulfilled.

> Execution defines the instructions. Environment defines and fulfils the physical and technological enablement required for those instructions to be executable as intended.

3. Environment

An Environment is a bounded physical or technological context established, provided or

accessed to enable enterprise instructions to be executed as intended.

An Environment may be:

· physical;
· technological;
· or a bounded combination of physical and technological enablement where separating them would obscure the actual enablement context.

Examples may include a workshop, office, warehouse, production environment, cloud environment, application environment, agent runtime environment, device environment or another bounded enablement context.

An Environment is not defined merely because a place or technology exists.

It becomes architecturally material where enterprise work requires it.

4. Environment Requirement

An Environment Requirement defines the physical or technological enablement that must be provided for enterprise instructions to be executable as intended.

The requirement belongs to Environment Architecture.

It may be identified because Execution exposes a need for enablement, but the physical or technological requirement itself is an Environment proposition.

For example, Execution may establish that work must:

· record a transaction;
· securely access information;
· lift a particular load;
· operate remotely;
· run an agent;
· preserve a controlled temperature;
· communicate with another system;
· or perform work in a safe physical setting.

Environment determines what physical or technological enablement is required to satisfy that need.

Execution exposes the need for enablement. Environment architects the Environment Requirement.

5. Requirement Fulfilment

Requirement Fulfilment defines how an Environment Requirement is satisfied.

Environment Architecture owns the fulfilment proposition because enablement is not fully architected merely by stating that something is required.

Fulfilment may involve:

· providing;
· building;
· configuring;
· acquiring;
· integrating;
· accessing;

· hosting;
· connecting;
· maintaining;
· or otherwise establishing the physical or technological enablement required.

The sourcing relationship through which fulfilment is obtained may involve Ecosystem Architecture.

The work required to establish or maintain fulfilment may involve Execution.

The people or agents participating may involve Workforce.

The actual condition of fulfilment is captured by Reality.

The enablement proposition itself remains Environment-owned.

> Environment owns the requirement and the fulfilment of that requirement.

6. Enabling Resource

An Enabling Resource is a physical or technological resource within or accessed through an Environment that contributes to fulfilment of an Environment Requirement.

Examples may include:

· buildings;
· workspaces;
· machinery;
· tools;
· equipment;
· vehicles;
· devices;
· networks;
· infrastructure;
· applications;
· platforms;
· databases;
· storage;
· compute;
· models;
· runtime components;
· integration mechanisms;
· security technologies;
· or other physical or technological resources.

This is not a closed taxonomy.

A resource does not become architecturally material merely because the enterprise owns or uses it.

Its materiality derives from the Environment Requirement it helps fulfil.

7. Requirement Traceability

Every material Environment Requirement should be traceable to the enterprise work or architectural condition that makes the enablement necessary.

The primary trace is usually to Execution.

The governing test is:

> Which instruction or executable work requirement would be impaired, prevented or materially changed if this enablement were unavailable?

The reverse test also applies:

> Which Environment Requirement establishes why this Enabling Resource is architecturally material?

These tests prevent Environment Architecture from becoming a generic asset inventory or technology catalogue.

8. Physical Enablement

Physical Enablement concerns the tangible conditions and resources required for work to be executable as intended.

It may include:

- premises;
- spatial arrangements;
- machinery;
- equipment;
- tools;
- vehicles;
- physical access mechanisms;
- utilities;
- storage;
- environmental controls;
- safety infrastructure;
- or another tangible enablement condition.

Environment owns the physical requirement and its fulfilment.

Execution owns the work performed within or through that physical context.

Reality captures actual physical condition, availability or state.

9. Technological Enablement

Technological Enablement concerns the digital, computational, information-processing and technical conditions required for work to be executable as intended.

It may include:

- applications;
- platforms;
- infrastructure;
- networks;
- integrations;
- devices;
- data stores;
- compute;
- models;
- agent runtimes;
- APIs;

· identity and access technologies;
· automation tooling;
· orchestration tooling;
· or another technological resource.

Technology is therefore enablement within AEA rather than a separate top-level architecture.

> Technology does not define the work. It enables the work.

10. Technology Enablement

Technology Enablement is a cross-architecture application proposition rather than a replacement name for Environment Architecture.

A technology-enabled operating mechanism may require:

· Environment for technological enablement;
· Ecosystem for providers and external relationships;
· Workforce for human, agent and organisational participation;
· Execution for work and instructions;
· Intelligence for information and understanding;
· Trust for reliance conditions;
· and other architectures as relevant.

Environment owns the technological environment requirement and fulfilment within that wider composition.

> Technology Enablement consumes Environment Architecture; it is not synonymous with the whole of Environment Architecture.

11. Environment Configuration

An Environment Configuration describes the designed arrangement of Enabling Resources and settings through which an Environment Requirement is intended to be fulfilled.

Configuration belongs to Environment where it describes how the enablement is architected.

The actual configuration present at a point in time belongs to Reality as Actuality.

> Environment owns intended configuration. Reality captures actual configuration.

A discrepancy between the two may indicate an enablement issue without transferring ownership of the Environment to Reality.

12. Environment Lifecycle

Environment Architecture may establish lifecycle propositions required to govern how an Environment or Enabling Resource should progress through relevant stages.

Lifecycle may concern matters such as:

· introduction;
· activation;
· operation;
· maintenance;
· upgrade;
· replacement;
· retirement;

· or another architecturally established stage.

No universal lifecycle taxonomy is imposed by this Published Standard.

Where a lifecycle model is established, Environment owns the meaning and intended progression of that model.

Reality captures the actual lifecycle state at a point in time.

> Lifecycle model belongs to Environment. Actual lifecycle state belongs to Reality.

13. Readiness

Environment Architecture may define Readiness as an intended enablement condition where work requires the Environment to meet specified criteria before execution can proceed as intended.

Environment owns:

· the meaning of readiness;
· the criteria;
· and the enablement requirement.

Reality captures whether those criteria are actually satisfied at the relevant time.

> Environment defines what ready means. Reality captures whether the Environment is actually ready.

Readiness must not be confused with Workforce readiness, Trust, or operational completion.

14. Availability

Where availability is a required characteristic of an Environment, Environment defines the availability requirement and its intended meaning.

Reality captures actual availability.

Intelligence may establish the information and derivation required to understand actual availability.

> Required availability is Environment. Actual availability is Reality.

The same ownership pattern applies to other Environment measures.

15. Environment Measures

Environment may define measures concerning whether enablement is being provided as required.

Examples may concern:

· availability;
· capacity;
· performance;
· configuration;
· resilience;
· readiness;
· utilisation;
· physical condition;
· technical condition;

· or another Environment proposition.
Environment owns the measure definition and intended condition.

Reality captures the actual observed or derived value.

Intelligence may own the derivation, evidence, provenance or information architecture required to establish the value.

Environment defines the measure !’ Intelligence may establish the derivation !’ Reality captures the actual value.

Measure ownership and measured Actuality are different propositions.

16. Access

An Environment may require technical or physical access controls in order for work to occur.

Environment owns the enablement mechanisms through which access is technically or physically provided.

Workforce Authority Structure owns binding enterprise decision rights.

Trust may use access, identity or authority conditions when determining warranted Reliance.

Technical access "` Authority.

A Participant being technically capable of performing an action does not establish that they are authorised to make the decision represented by that action.

Likewise, a system permission does not itself establish Trust.

17. Identity and Access Technology

Identity and access technologies may form part of Environment Architecture where they provide technological enablement.

Examples may include authentication, access-control systems, credentials infrastructure or technical identity services.

The technology that enforces access is Environment.

The Workforce Authority proposition determining whether a human may make a binding decision remains Workforce.

The Trust proposition concerning whether a party can be relied upon remains Trust.

The Ecosystem or Workforce identity of the Actor or Participant remains owned by the relevant source architecture.

Technology can enforce a boundary without owning the authority, identity or reliance proposition that gives the boundary meaning.

18. Agent Runtime

An agent runtime is technological enablement through which an agent can execute work.

The runtime belongs to Environment Architecture.

The agent's Role and Participant identity belong to Workforce Architecture.

The instructions executed by the agent belong to Execution Architecture.

The information and organisational understanding required by the agent are supported through

Intelligence Architecture.

Reliance conditions applicable to the agent belong to Trust Architecture.

Actual agent performance belongs to Reality.

> Runtime is Environment. Agent participation is Workforce. Instructions are Execution.

This distinction prevents technology configuration from becoming a substitute for workforce or process architecture.

19. Automation and Orchestration

Automation and orchestration mechanisms are forms of technological enablement where they provide the runtime or technical mechanism through which work is executed, routed or coordinated.

They do not constitute additional Workforce sourcing routes.

They do not independently define the Process.

They do not create Authority.

Execution determines the work and instructions.

Workforce determines participation.

Environment provides the automation or orchestration mechanism.

> Automation changes fulfilment mechanics; it does not erase architectural ownership.

20. Environment and Execution Architecture

Execution is the primary architectural anchor for Environment Requirements.

Execution defines the enterprise's instructions for how work is performed.

Environment defines and fulfils the physical and technological enablement required for those instructions to be executable as intended.

The governing handoff is:

> Execution Instruction !’ Environment Requirement !’ Requirement Fulfilment

This does not mean Execution owns the Environment Requirement.

Execution exposes the executable need.

Environment translates that need into a physical or technological enablement proposition.

A change in Environment does not automatically create an Execution Variant.

Where the Environment change materially changes executable work while preserving the Canonical Invariant Contract, Execution may establish an Approved Variant.

Where executable work remains materially unchanged, no Variant is required merely because the technology, device, location or other enablement changed.

21. Environment and Workforce Architecture

Workforce owns Roles, Participants, Skills Required, Sourcing and Workforce Structures.

Environment owns the physical and technological enablement required for the work through which participation occurs.

A Workforce proposition may reveal characteristics that Environment must accommodate.

For example, remote participation, agent participation or specialised human participation may expose different enablement needs.

Environment determines the resulting physical or technological requirement and fulfilment.

Workforce defines participation. Environment defines the physical and technological enablement required for the work through which that participation occurs.

Technical access does not confer Workforce Authority.

Environment must not use system permissions as a substitute for the Authority Structure.

22. Environment and Ecosystem Architecture

An external provider may be an Ecosystem Actor because the enterprise holds a material Relationship with it.

The physical or technological enablement obtained through that Relationship remains Environment Architecture.

For example:

- a cloud provider is an Ecosystem Actor;
- the provider contract is an Ecosystem Relationship Instrument;
- the cloud environment used by the enterprise is Environment;
- actual service availability is Reality;
- reliance upon the provider is Trust.

Ecosystem owns the provider Relationship. Environment owns the enablement requirement and fulfilment.

The provider's technology does not become Ecosystem-owned merely because it is externally supplied.

23. Environment and Domain Architecture

Domain owns Business Models, Value Chains and Capabilities.

Environment may provide enablement required for work supporting those Capabilities.

A system is not a Capability.

A building is not a Capability.

A platform is not a Capability.

An agent runtime is not a Capability.

Capability describes a stable business ability. Environment provides physical and technological enablement through which work supporting that ability can be performed.

A change to an Enabling Resource does not automatically change Capability identity.

24. Environment and Intelligence Architecture

Environment may provide the systems, platforms, repositories, compute and other technological enablement through which Intelligence operates.

Intelligence owns the architecture of organisational knowing and Understanding.

Environment does not acquire ownership of information meaning, evidence, provenance,

confidence or Organisational Memory merely because those things are stored or processed in technology.

Likewise, Intelligence does not own the technological environment merely because it depends upon it.

> Environment enables Intelligence. Intelligence establishes Understanding.

Where Environment itself requires understanding of configuration, capacity, availability or other Environment propositions, Environment originates the Requirement of Understanding and Intelligence architects its fulfilment.

25. Environment and Reality Architecture

Environment owns:

- Environment identity;
- Environment Requirements;
- Requirement Fulfilment;
- Enabling Resources;
- intended Configuration;
- lifecycle models;
- readiness definitions;
- and Environment measure definitions.

Reality captures actual:

- availability;
- configuration;
- condition;
- capacity;
- lifecycle state;
- readiness;
- performance;
- fulfilment;
- and other Environment Actuality.

> State is not identity.

Reality may capture that an Environment Requirement is not currently fulfilled.

That does not transfer the Environment Requirement to Reality.

> Environment defines what enablement should exist. Reality captures what enablement actually exists.

26. Environment and Trust Architecture

Trust may depend upon Environment propositions as Reliance Conditions.

Examples may include:

- security controls;
- access mechanisms;
- availability;
- resilience;
- isolation;
- technical safeguards;

· physical controls;
· or another enablement condition.

Environment owns the underlying physical or technological proposition.

Reality captures its actual state.

Trust determines the reliance consequence.

Environment establishes the enablement condition. Reality captures whether it actually holds. Trust determines whether reliance is warranted.

A secure technical configuration does not establish universal Trust.

A trusted Participant does not eliminate the need for appropriate Environment controls.

27. Environment and Directional Architecture

Directional choices may create implications for Environment.

A strategic choice to operate digitally, enter a physical market, adopt a distribution model, automate work or scale through a particular mechanism may create new enablement requirements.

Directional owns the strategic choice.

Environment owns the resulting physical and technological enablement architecture.

Direction determines what the enterprise chooses to realise. Environment determines the physical and technological enablement required by the adopted work.

Environment does not independently make the strategic choice because one technology is available.

28. Environment and Possibility Architecture

Possibility may explore future Environment configurations, technologies, facilities, platforms, runtimes or other enablement options.

These remain Possibility propositions until adopted.

Environment owns the established requirement and intended fulfilment.

Possibility owns the hypothesis or option concerning what that Environment could become.

Possibility explores future enablement. Environment defines adopted enablement.

The existence of a new technology does not automatically create an Environment Requirement.

29. Environment and Expression Architecture

Physical and technological Environments may carry enterprise expression.

A retail space, website, application, interface or device may be a medium through which Enterprise Identity is encountered.

Expression owns what the manifestation should express and the intended experience.

Environment owns the physical or technological enablement through which that expression is made possible.

Environment enables the manifestation. Expression defines what the manifestation should express.

The actual experience remains Reality.

30. Environment and AEA Governance

Environment Architecture may define governed enablement requirements and fulfilment propositions.

AEA Governance controls the authority, lifecycle, version, change, conformance, release and archival treatment of AEA Standards and governed architecture material.

Technology administration rights do not confer standards governance authority.

Likewise, operational ownership of a system does not automatically confer authority to alter the architectural proposition the system fulfils.

> Administrative control of technology is not architectural authority.

31. Environment and AEA in Action

AEA in Action may combine Environment with other architectures to construct technology enablement models, operating mechanisms, implementation views, BOAS, SOPs, runtime designs and other purpose-specific renders.

Environment contributes:

- Environment Requirements;
- Requirement Fulfilment;
- Enabling Resources;
- intended Configuration;
- and other relevant enablement propositions.

Execution contributes work.

Workforce contributes participation.

Ecosystem contributes providers and relationships.

Intelligence contributes Understanding.

Trust contributes Reliance Conditions.

AEA in Action performs the cross-architecture composition.

Environment does not absorb the whole technology-enabled operating model merely because technology is prominent within it.

32. Culture Boundary

Culture is not an Environment Architecture object.

Physical and technological conditions can materially influence lived culture.

They may shape how people communicate, collaborate, access information, experience work or participate in the enterprise.

But culture emerges from the wider Architecture Set.

Expression may contribute internal identity expression.

Workforce contributes participation and structures.

Execution contributes work.

Trust contributes Reliance.

Reality captures lived Actuality.

Environment may shape cultural conditions. It does not own culture.

33. Governing Principles

Environment Architecture is the architecture of Enablement.

Environment Architecture defines the physical and technological enablement required to execute the enterprise's instructions for delivering its intended outcomes.

Execution exposes the need for enablement. Environment architects the Environment Requirement.

Environment owns the requirement and the fulfilment of that requirement.

An Enabling Resource is architecturally material because of the Environment Requirement it fulfils.

Environment Architecture is not a generic asset inventory or technology catalogue.

Technology does not define the work. It enables the work.

Technology Enablement consumes Environment Architecture; it is not synonymous with the whole of Environment Architecture.

Environment owns intended Configuration. Reality captures actual Configuration.

Lifecycle model belongs to Environment. Actual lifecycle state belongs to Reality.

Environment defines what ready means. Reality captures whether the Environment is actually ready.

Required availability is Environment. Actual availability is Reality.

Environment defines the measure !’ Intelligence may establish the derivation !’ Reality captures the actual value.

Technical access "` Authority.

Technology can enforce a boundary without owning the authority, identity or reliance proposition that gives the boundary meaning.

Runtime is Environment. Agent participation is Workforce. Instructions are Execution.

Automation and orchestration are enablement mechanisms, not additional Workforce sourcing routes.

A change in Environment creates an Execution Variant only where executable work materially changes while Process Module identity remains invariant.

Ecosystem owns the provider Relationship. Environment owns the enablement requirement and fulfilment.

Environment enables Intelligence. Intelligence establishes Understanding.

Environment establishes the enablement condition. Reality captures whether it actually holds. Trust determines whether reliance is warranted.

Environment may shape cultural conditions. It does not own culture.

34. AI Protocol

AI may assist Environment Architecture by:

· identifying candidate Environment Requirements from established Execution instructions;
· distinguishing physical from technological enablement;

- identifying Enabling Resources;
- testing whether a resource has a traceable Environment Requirement;
- identifying candidate Requirement Fulfilment options;
- comparing intended and actual Configuration where Reality is available;
- identifying lifecycle or readiness implications;
- identifying technology dependencies;
- identifying where technical access has been mistaken for Authority;
- identifying where an agent runtime has been mistaken for the agent Role;
- identifying where automation or orchestration has been mistaken for Workforce sourcing;
- and detecting Environment changes that may require Execution impact assessment.

AI must not:

- create Environment Requirements merely because technology exists;
- treat every enterprise asset as an architecturally material Enabling Resource;
- treat technology as a Capability;
- treat a system as a Process;
- treat an agent runtime as the agent Participant;
- treat automation as a Workforce sourcing route;
- infer Authority from technical access;
- infer Trust from a security configuration;
- treat intended Configuration as actual Configuration;
- treat a lifecycle model as actual lifecycle state;
- treat readiness criteria as proof of actual readiness;
- treat required availability as actual availability;
- create an Execution Variant merely because the Environment changed;
- infer a strategic technology choice from current tooling;
- treat provider technology as Ecosystem-owned merely because the provider is external;
- or expose proprietary AEA in Action composition methods as Published Standard doctrine.

AI may assist the architecture of enablement. It must not manufacture requirements, actuality, authority or architectural ownership from technology.

35. Conformance

A conformant application of Environment Architecture must preserve, where applicable:

1. the relevant Environment; 2. the Environment Requirement; 3. Requirement Fulfilment; 4. Enabling Resources; 5. traceability to the work or architectural condition creating the need for enablement; 6. intended Configuration where material; 7. lifecycle model where material; 8. readiness definition where material; 9. Environment measure definitions where material; 10. distinction between intended and actual Configuration; 11. distinction between lifecycle model and actual lifecycle state; 12. distinction between readiness criteria and actual readiness; 13. distinction between required and actual availability; 14. distinction between technical access and Workforce Authority; 15. distinction between agent runtime and Workforce Participant; 16. distinction between automation/orchestration and Workforce sourcing; 17. distinction between provider Relationship and provided enablement; 18. Requirements of Understanding originating in Environment where applicable; 19. distinction between Environment propositions and Reality Actuality; 20. and cross-architecture ownership where Environment contributes to wider operating mechanisms.

36. Non-Conformance

An application is non-conformant where it:

· reduces Environment Architecture to an asset register;
· reduces Environment Architecture to an application portfolio;
· treats every technology as architecturally material without an Environment Requirement;
· treats a system as a Capability;
· treats a technology as a Process;
· assigns Environment Requirement ownership to Execution;
· states a requirement without architecting its fulfilment where fulfilment is required;
· treats actual configuration as Environment-owned Actuality;
· treats lifecycle state as Environment-owned Actuality;
· treats actual readiness as Environment-owned Actuality;
· treats actual availability as Environment-owned Actuality;
· treats technical access as Authority;
· treats security controls as proof of Trust;
· treats an agent runtime as a Workforce Role or Participant;
· treats automation or orchestration as a Workforce sourcing route;
· creates an Execution Variant merely because technology or location changed;
· treats an external provider's technology as an Ecosystem object rather than distinguishing provider Relationship from enablement;
· treats culture as Environment-owned;
· or treats Technology Enablement as synonymous with Environment Architecture.

37. Published Gaps

GAPS — None currently identified that prevent publication of this Standard.

Detailed Environment Requirement fields, Requirement Fulfilment schemas, Enabling Resource classifications, Configuration models, lifecycle models, readiness criteria, measure schemas, physical and technological reference patterns and native schematic construction protocols belong in the downstream Environment Architecture package.

Their absence from this Published Standard does not constitute an architecture doctrine gap.

38. Publication Position

Environment Architecture provides AEA with its architecture of Enablement.

It begins not with the assets the enterprise happens to own, nor with the technologies currently available, but with the physical and technological conditions required for enterprise instructions to be executable as intended.

Its governing chain is:

Execution Instruction !’ Environment Requirement !’ Requirement Fulfilment !’ Enabling Resource(s)

Environment owns the requirement and the intended fulfilment.

Reality captures whether that enablement actually exists and in what condition.

This preserves the critical distinctions:

required enablement "` actual enablement

intended Configuration "` actual Configuration

lifecycle model "` actual lifecycle state

readiness definition "` actual readiness

required availability "` actual availability

technical access "` Authority

runtime "` Participant

automation "` Workforce sourcing

provider Relationship "` provided Environment

and:

technology "` the work it enables.

Environment can therefore evolve without silently redefining the enterprise's Capabilities, Processes, Roles or Trust propositions.

The governing proposition remains:

Environment Architecture defines the physical and technological enablement required to execute the enterprise's instructions for delivering its intended outcomes, and owns both the requirements for that enablement and the fulfilment through which those requirements are satisfied.